LA-UR-98-5659

Title: TRACER-PARTICLE ADVECTION: ALGORITHM COMPONENTS AND IMPLEMENTATION METHODS

Author(s) Jerry S. Brock, X-CI
James W. Painter, X-CI
Douglas B. Kothe, MST-8

Photograph by Chris J. Lindberg

Los Alamos
NATIONAL LABORATORY

Form 836 (10/96)

Tracer-Particle Advection:
Algorithm Components and Implementation Methods

Jerry S. Brock, James W. Painter, and Douglas B. Kothe

Abstract

Tracer-particles, massless particles that are advected throughout the flow domain, are an important factor in computational solutions. The utility of tracer-particles, which sample and report relevant solution data, is predicated upon an accurate advection algorithm. The objective of this research was to investigate the ability of existing techniques, physical-space and logical-space advection, to accurately predict tracer-particle pathlines within various grid topologies. The physical-space technique, using a higher-order integration method, accurately predicted pathlines for a curved flow-field within a nonorthogonal grid. In contrast, the logical-space technique failed to accurately predict pathlines for a uniform flow-field within a nonuniform rectilinear grid. Existing logical-space advection techniques are, therefore, limited to uniform rectilinear grids.

Introduction

A primary goal of computational simulation research is the development of accurate and efficient methods to solve physical models. The full power and utility of simulation codes requires the ancillary ability to perform insightful data interpretation [1]. Data interpretation is used to examine the solution and also as a diagnostic tool to improve the computational technique [2]. The purpose of this research was to investigate tracer-particle methods which are used extensively for data extraction within material advection models. Material advection, the forced motion of solid or fluid material, is a fundamental physical process and is the core of many simulation codes. Tracer-particles, massless particles that are advected throughout the computational domain concurrent with the solution evolution, are used to sample and report relevant data. The utility of tracer-particles is predicated upon an accurate and efficient advection algorithm [3].

Some popular post-processing tools [4] generate steady-state streamlines, which obviates a tracer-particle capability within time-independent simulation codes. For time-dependent simulations, some post-processing tools [7] generate unsteady tracer-particle pathlines by interpolating velocity flow-field data between solution time-steps. However, this introduces addition numerical approximations within the advection algorithm, and it may reduce the accuracy of the tracer-particle pathline. Therefore, reliance on a post-processing tool for accurate unsteady pathline evolution is questionable. In addition, the exclusion of a pathline capability within the simulation code limits the utility of tracer-particles as a model diagnostic tool. A tracer-particle capability is then recommended for inclusion within all modern simulation codes.

Two types of tracer-particle advection are recognized: physical-space and logical-space

advection. Several physical-space advection techniques appear in the literature [5-9]. However, the implementation details and validation of these algorithms is often poorly documented; in general, tracer-particle advection is relatively simple compared to the physical models solved in modern simulation codes. The lack of documentation is especially true for logical-space advection techniques which are used within legacy codes [10]. Where logical-space advection techniques do appear in the literature [11], their results are not adequately investigated. In fact, these techniques produce surprisingly poor results even for simple flow conditions computed on common grid systems [12]. Therefore, prior to using tracer-particle data, it is important to understand the capabilities and limitations of their deceptively simple advection algorithm.

The objective of this research was to investigate the ability of existing tracer-particle advection techniques to accurately predict pathlines within various grid topologies. The objective was also to describe the advection algorithm as a collection of fundamental, yet independent components and to present some of their common methods of implementation. The component description of the algorithm matches the object-oriented paradigm inherent in advanced programming languages and also adopted by modern simulation codes. This modular description can produce many unique advection techniques; the specific choice of implementation method for one component does not restricted the selections made for the other portions of the algorithm.

This article continues by presenting the fundamental components of the tracer-particle advection algorithm: the advection equation, integration method, localization method, and the Jacobian matrix. The advection equation determines which spatial coordinates are advected, and the integration method determines the evolution of the tracer-particle pathline. The localization method maintains the particle-to-grid connectivity data which is required to interpolate the advection velocity. The implementation methods for two algorithm components, the advection equation and the localization method, are divided into physical-space and logical-space categories. Some popular choices for the Jacobian matrix, which describes the transformation between the physical-space and logical-space coordinate systems, will also be presented.

Following the description of the tracer-particle algorithm, two test problems are presented: physical-space and logical-space advection. Each test problem represents a different, yet common method of computing tracer-particle pathlines. The physical-space test problem focuses on the accuracy of the integration methods, and the various Jacobian matrices are examined in the logical-space test problem. In this study, the capability and limitation of each advection technique is examined particularly with respect to various grid topologies such as uniform or nonuniform grids and orthogonal or nonorthogonal grids. Following the presentation and discussion of the two test problems, a summary of this research concludes this article.

Advection Equation

A tracer-particle's position may be defined within either a physical-space or logical-space coordinate system. The time evolution of the tracer-particle's position, the pathline or trajectory, may be computed within either of these reference frames. The advection equation defines which of the coordinates, physical-space, $\bar{X}$, or logical-space, $\bar{\xi}$, are advanced in time to establish the pathline. The pathline may then be determined in the other reference frame with a known coordinate system transformation. The choice of advection equation has a large impact on the accuracy and computational efficiency of the resulting tracer-particle advection technique.

Physical-Space Advection

A tracer-particle's position may be described within three-dimensional physical space as $\bar{X} = \bar{X}(x, y, z)$. This vector represents the tracer-particle's location within a global reference frame. The i-th element of this coordinate vector is noted as $\bar{X}_i$, where i varies from one to ndim, the number of spatial dimensions. The physical-space advection equation simply defines the tracer-particle's physical velocity, $\bar{V}_x(\bar{X}(\bar{\xi}), t)$, as presented in Equation 1.

$$\frac{\partial \bar{X}}{\partial t} = \bar{V}_x(\bar{X}(\bar{\xi}), t) \tag{1}$$

Logical-Space Advection

A tracer-particle's position may also be described within three-dimensional logical space as $\bar{\xi} = \bar{\xi}(\xi, \eta, \zeta)$. If this vector represents the tracer-particle's location within a computational cell and each coordinate is bounded by zero and unity, $\bar{\xi}$ represents the local logical coordinates. The Jacobian matrix, $J = \partial \bar{X}(\bar{\xi}) / \partial \bar{\xi}$, is then used to relate the physical velocity to its counterpart, the logical velocity, $\bar{V}_\xi(\bar{X}(\bar{\xi}), t)$. The logical-space advection equation simply defines the tracer-particle's velocity within the logical reference frame as presented in Equation 2.

$$\frac{\partial \bar{\xi}}{\partial t} = \left[\frac{\partial \bar{X}(\bar{\xi})}{\partial \bar{\xi}} \right]^{-1} \bar{V}_x(\bar{X}(\bar{\xi}), t) = \bar{V}_\xi(\bar{X}(\bar{\xi}), t) \tag{2}$$

The logical-space advection equation may also be used to define tracer-particle pathlines within a global reference frame. Existing logical-space advection techniques [10,11] utilize this alternative strategy because the localization process, detailed later in this article, is easily implemented within structured grid systems. However, the local logical coordinates are always required for data interpolation. More importantly, these coordinates are more generally suitable for use within unstructured grid systems where a simple conversion between global and local logical coordinates is not guaranteed. Therefore, this research only investigated tracer-particle logical-space advection within a local, cell-based reference frame.

The physical-space and logical-space advection equations are both concise statements which simply define a tracer-particle equation-of-motion within their respective reference frames. Both of these equations are also functions of the physical velocity, $\bar{V}_x(\bar{X}(\bar{\xi}), t)$. The physical velocity is presented as a function of logical coordinates to convey the necessary interpolation from the grid to the tracer-particle. Interpolation requires particle-to-grid connectivity data which includes a cell number, $Cell$, and cell-based, local logical coordinates, $\bar{\xi}$. The popular trilinear interpolation method, which simplifies into bilinear and linear versions, was used in this research.

Integration Method

Tracer-particle pathlines are obtained by integrating either the physical-space or logical-space advection equations. Once a discrete velocity flow-field has been selected for each time-step, the equations-of-motion become ordinary differential equations. Euler integration methods represent a popular solution strategy for the resulting initial value problem. These methods are relatively efficient, requiring only a limited number of forcing-function evaluations per time-step. They may also provide sufficiently accurate, second-order solutions; material advection solutions are typically second-order accurate. This research used two common integration methods: the standard Euler method and the modified Euler, predictor-corrector method [9, 13].

Standard Euler Method

The standard Euler integration method requires a single forcing function evaluation, and produces a first-order accurate solution. For the physical-space advection example which follows in Equation 3, the forcing function is simply the physical velocity. During a single time-step, Δt, the tracer-particle's position is advanced from time 't' to time '$t + \Delta t$'. The corresponding discrete times are time-level 'n' and time-level '$n + 1$' respectively. The advecting velocity is evaluated at the initial, time-level 'n' position: $\bar{V}_x^n = \bar{V}_x(\bar{X}(\bar{\xi}^n))$. The standard Euler method applied to the physical-space tracer-particle advection equation is presented in Equation 3.

$$\bar{X}^{n+1} = \bar{X}^n + \bar{V}_x^n \bullet \Delta t \tag{3}$$

Predictor-Corrector Method

The modified Euler, predictor-corrector integration method requires two forcing function evaluations per time-step, and produces a second-order accurate solution. For the logical-space advection example which follows in Equation 4, the forcing function is the logical velocity. The logical velocity vector is a product of the inverse Jacobian matrix and the physical velocity vector: $\bar{V}_\xi = [J]^{-1}\bar{V}_x$. This popular integration method evaluates one forcing function at the initial position and another at the time-level '$n + 1/2$' position. The corrector velocity is evaluated at this intermediate position: $\bar{V}_\xi^{n+1/2} = \bar{V}_\xi(\bar{X}(\bar{\xi}^{n+1/2}))$. The modified Euler, predictor-corrector

method applied to the logical-space tracer-particle advection equation is presented in Equation 4.

$$\begin{aligned} \bar{\xi}^{n+1/2} &= \bar{\xi}^{n} + \bar{V}_{\xi}^{n} \bullet \Delta t/2 \\ \bar{\xi}^{n+1} &= \bar{\xi}^{n} + \bar{V}_{\xi}^{n+1/2} \bullet \Delta t \end{aligned} \tag{4}$$

At the conclusion of each time-step, a new tracer-particle position vector is obtained: either $\bar{X}^{n+1}$ or $\bar{\xi}^{n+1}$. A new set of particle-to-grid connectivity data, used for data interpolation, is also required prior to the next time-step: cell number, $Cell^{n+1}$, and logical coordinates, $\bar{\xi}^{n+1}$. If physical-space advection was selected, new logical-coordinates must also be computed at the end of each time-step. In contrast, if logical-space advection was selected, no additional work is required; physical-space coordinates, which may be necessary for data output, are not required for logical-space advection. This reduction in computational effort is one indication that the logical-space advection technique may be preferred for computing tracer-particle pathlines.

Localization Method

A tracer-particle's position within any computational grid is properly defined by a cell number, $Cell$, and bounded logical coordinates, $0 < \bar{\xi} < 1$, relative to that cell. This particle-to-grid connectivity data is required to accurately interpolate the discrete velocity field, typically defined at cell vertices, to the tracer-particle position. At the end of each time-step, new logical coordinates, $\bar{\xi}^{n+1}$, are established relative to $Cell^{n}$. If each component of this position vector is bounded, then $Cell^{n}$ becomes the new cell number, $Cell^{n+1}$. However, if any element of $\bar{\xi}^{n+1}$ is unbounded, either $\bar{\xi}_i^{n+1} < 0$ or $\bar{\xi}_i^{n+1} > 1$ relative to $Cell^{n}$, then the tracer-particle has changed cells and the particle-to-grid connectivity data is incompatible. A search or localization process is then required to reestablish the proper particle-to-grid connectivity information.

Regardless of whether the physical-space or logical-space equation was selected for advection, a localization process is necessary when a tracer-particle changes cells. For multi-step integration methods, such as the modified Euler predictor-corrector method, this search process may be required multiple times during each time-step. The tracer-particle localization method used in this research, similar to those found in the literature [14-18], has three basic components which are implemented in sequence: 1) guess a new cell number, $Cell^{n+1}$, 2) evaluate new logical coordinates, $\bar{\xi}^{n+1}$, relative to $Cell^{n+1}$, and 3) test for cell and logical coordinate compatibility, $0 < \bar{\xi}^{n+1} < 1$. Both the first and second component of this process possess two unique methods of implementation: logical-space and physical-space methods.

Cell Number

Logical-Space Method The objective of the logical-space method is to predict a new cell number while only using the existing incompatible particle-to-grid connectivity data: Cell^n and $\bar{\xi}^{n+1}$. For structured grid systems, the existing cell number may be translated into a coordinate index vector, $\bar{I}^n = \bar{I}^n(i, j, k)$. A new index vector may then be predicted by analyzing each logical coordinate; $\bar{I}_i^n$ is decremented by one if $\bar{\xi}_i^{n+1} < 0$, and $\bar{I}_i^n$ is incremented by one if $\bar{\xi}_i^{n+1} > 1$. The modified index vector, $\bar{I}_i^{n+1}$, may then be translated into a new cell number, Cell^{n+1}. For orthogonal grids, this method only requires one estimate of the new cell number. In contrast, the resulting localization process is iterative if the grid is nonorthogonal.

An alternative strategy for logical-space advection was previously described: computing tracer-particle pathlines by advecting their global logical coordinates, $\bar{\xi}_g$. When this strategy is utilized within a uniform structured grid system, the computational expense of the advection algorithm is reduced because a simple and deterministic localization process may be employed. The integer portion of $\bar{\xi}_g$ identifies the cell number, $\mathrm{Cell} = \mathrm{int}(\bar{\xi}_g)$, and the fractional portion defines the local logical coordinates, $\bar{\xi}_l = \bar{\xi}_g - \mathrm{int}(\bar{\xi}_g)$. This reduction in computational expense is another indication that the logical-space advection technique may be preferred for computing tracer-particle pathlines. However, the ability to accurately predict pathlines, addressed later in this article, is the only valid criterion for selecting an advection technique.

Physical-Space Method The objective of the physical-space method is to predict a new cell number using Cell^n and the tracer-particle's latest physical coordinates. If physical-space advection was selected then $\bar{X}^{n+1}$ is available, otherwise it must be interpolated using the existing incompatible particle-to-grid connectivity data. A list of cells neighboring Cell^n may then be created and appropriately sorted. If the most probable new cell number is stored first, the list may be sampled in descending order to predict Cell^{n+1}. For structured grid systems, the number of cell neighbors is constant throughout the domain. In contrast, for unstructured grid systems there may be a unique number of neighbors for each cell in the domain.

The physical-space method of predicting new cell numbers is applicable within orthogonal and nonorthogonal grids. However, the resulting localization process may be deterministic or iterative, depending on the quality of the sorting technique. One non-directional sorting technique would determine the nearest neighbor cell-centers relative to the tracer-particle position. A directional sorting technique might align the tracer-particle velocity vector with a particle-to-cell-center distance vector. Each of these methods, utilized in Reference [19], represents a practical technique for sorting neighboring cells within a localization process. However, the most efficient sorting method may be grid dependent, and its study was beyond the scope of this research.

Logical Coordinates

Logical-Space Method At the second stage of the localization process, the existing particle-to-grid connectivity data includes the new cell number, Cell^{n+1}, and the old logical coordinates, $\bar{\xi}_{\mathrm{old}}^{n+1}$, which are defined relative to Cell^{n}. New logical coordinates, $\bar{\xi}_{\mathrm{new}}^{n+1}$, which are defined relative to Cell^{n+1} are required. The objective of the logical-space method is to evaluate $\bar{\xi}_{\mathrm{new}}^{n+1}$ using only the available particle-to-grid connectivity data. For structured grid systems, the logical-space method of evaluating new logical coordinates is presented in Equation 5, where $\bar{a}_i = 1$ if $\bar{\xi}_{\mathrm{old},\,i}^{n+1} > 1$ else $\bar{a}_i = 0$, and $\bar{b}_i = 1$ if $\bar{\xi}_{\mathrm{old},\,i}^{n+1} < 0$ else $\bar{b}_i = 0$.

$$\bar{\xi}_{\mathrm{new}}^{n+1} = \mathrm{J}_{\mathrm{new}}^{-1}\mathrm{J}_{\mathrm{old}}(\bar{\xi}_{\mathrm{old}}^{n+1} - \bar{a}) + \bar{b} \tag{5}$$

The Jacobian matrices in Equation 5, $\mathrm{J}_{\mathrm{old}}$ and $\mathrm{J}_{\mathrm{new}}$, account for the difference in spatial coordinate transformations between the old and new tracer-particle cells. If these matrices are functions of $\bar{\xi}$, $\mathrm{J} = \mathrm{J}(\bar{\mathrm{X}}(\bar{\xi}))$, then an iterative solution of Equation 5 is required. In contrast, if these matrices are not functions of $\bar{\xi}$, this method only requires a single estimate for new logical coordinates. Furthermore, within uniform orthogonal grid systems, the Jacobian matrices are identical for every cell, and the product $\mathrm{J}_{\mathrm{new}}^{-1}\mathrm{J}_{\mathrm{old}}$ reduces to the identity matrix. Various methods of constructing the Jacobian matrix are discussed in the following section of this article.

Physical-Space Method The objective of the physical-space method is to evaluate new logical coordinates, $\bar{\xi}_{\mathrm{new}}^{n+1}$, defined relative to Cell^{n+1}, using the tracer-particle's latest physical position, $\bar{\mathrm{X}}^{n+1} = \bar{\mathrm{X}}_{\mathrm{p}}(\bar{\xi}_{\mathrm{p}})$. If physical-space advection was selected then $\bar{\mathrm{X}}^{n+1}$ is available, otherwise it must be interpolated with the existing incompatible particle-to-grid connectivity data: Cell^{n} and $\bar{\xi}_{\mathrm{old}}^{n+1}$. One method employs a Taylor-series expansion defined from the cell origin to the tracer-particle's physical position [5]. If the higher-order expansion terms are neglected, the resulting system of equations may be solved iteratively to determine a logical coordinate vector. The physical-space method of evaluating logical coordinates is presented in Equation 6.

$$\left[\frac{\partial\bar{\mathrm{X}}(\bar{\xi}^{\mathrm{i}})}{\partial\bar{\xi}}\right]\{\delta\bar{\xi}\} = -\{\bar{\mathrm{X}}(\bar{\xi}^{\mathrm{i}}) - \bar{\mathrm{X}}_{\mathrm{p}}(\bar{\xi}_{\mathrm{p}})\} \tag{6}$$

Within Equation 6, the logical-coordinate delta-vector is defined as $\delta\bar{\xi} = \bar{\xi}^{\mathrm{i}+1} - \bar{\xi}^{\mathrm{i}}$, where the superscript i is the iteration index. In this research, a convergence tolerance for $\delta\bar{\xi}$ was set to 10^{-6}, and the initial guess of the logical-coordinates, $\bar{\xi}^{\mathrm{i}=1}$, was specified as the null vector. Note that, if the coordinate transformation matrix is constant, then only a single solution of Equation 6 is required to compute a logical coordinate vector.

The above methods of evaluating logical coordinates are straightforward, however, they represent the most costly portion of tracer-particle advection. Equations 5 and 6, which may require an iterative solution, affect the advection algorithm in three ways. First, if physical-space advection was selected, then after each time-step when $\bar{X}^{n+1}$ is computed, $\bar{\xi}^{n+1}$ must also be evaluated. Second, when a tracer-particle changes cells, many converged solutions of Equations 5 or 6 may be required to satisfy a single search process. Third, if the predictor-corrector integration method is used to evolve the pathline, two localization solutions may be required for each time-step. Therefore, Equations 5 and 6 are key factors in the efficiency of tracer-particle advection.

Jacobian Matrix

The Jacobian matrix, $J = \partial\bar{X}(\bar{\xi})/\partial\bar{\xi}$, is another key element of tracer-particle advection. This coordinate transformation matrix appears within both localization methods described above. The Jacobian matrix affects the iterative solution and, thus, the computational expense of solving Equations 5 and 6 for logical coordinates. However, as a portion of the search method, Equations 5 and 6 do not affect the accuracy of the resulting tracer-particle pathline.

More importantly, the Jacobian matrix affects the accuracy of logical-space advection; this matrix transforms the physical velocity into a logical-space equation-of-motion. Within nonuniform and nonorthogonal grids, the coordinate transformation varies discretely between cells. To mitigate the effects of these discontinuities within the integration method, existing logical-space advection techniques use various Jacobian matrices. The objective of this research was to investigate these and other simple coordinate transformations. Methods for evaluating the Jacobian matrix are generally divided into methods for orthogonal and nonorthogonal grid.

Orthogonal Grids

Three-dimensional computational space is often discretized into orthogonal, hexahedral cells. If the straight edges of these cells are aligned with the physical reference frame, the grid is rectilinear. Computational cells within a rectilinear grid generally arranged into a structured system and numbered with a coordinate index vector, $\bar{I} = \bar{I}(i, j, k)$. More importantly, for rectilinear grids the physical coordinates are uncoupled within the spatial transformation and the Jacobian matrix is diagonal. The matrix diagonal may be expressed as a vector of finite-differences associated with the tracer-particle position, $\Delta\bar{X}_p$, if $\partial\bar{\xi} \approx \Delta\bar{\xi} = 1$. The resulting diagonal Jacobian matrix for three-dimensional space is presented in Equation 7.

$$J = \begin{bmatrix} \Delta x_p & 0 & 0 \\ 0 & \Delta y_p & 0 \\ 0 & 0 & \Delta z_p \end{bmatrix} \tag{7}$$

Cell-Based Methods The objective of cell-based methods to describe the Jacobian matrix is to evaluate the coordinate transformation using only computational cell information. For rectilinear grids, one method approximates the Jacobian matrix diagonal vector as the tracer-particle cell widths, $\Delta\overline{X}^{c}_{\bar{i}}$. The resulting Jacobian matrix is constant throughout the cell, $J \neq f(\bar{\xi})$. The cell-based, piece-wise constant (Cell: Pwc) Jacobian matrix is presented in Equation 8.

$$\Delta\overline{X}_{p} = \Delta\overline{X}^{c}_{\bar{i}} \tag{8}$$

Another cell-based technique for evaluating coordinate transformations within rectilinear grids is a piece-wise linear method. Within nonuniform grids, where $\Delta\overline{X}^{c}$ varies, the cell-based, piece-wise constant Jacobian matrix is discontinuous across cells. One method of smoothing this out is to linearly interpolate tracer-particle cell widths between cell centers, $\overline{X}^{cc}$. The resulting Jacobian matrix is implicitly a function of the tracer-particle's logical coordinates, $J = J(\overline{X}(\bar{\xi}))$. The cell-based, piece-wise linear (Cell: Pwl) Jacobian matrix is presented in Equation 9.

$$\begin{aligned}
\bar{\xi}_{p} < 1/2 \;\; ; \;\; \Delta\overline{X}_{P} &= \Delta\overline{X}^{c}_{\bar{i}-1} + \left(\frac{\overline{X}_{p}(\bar{\xi}_{p}) - \overline{X}^{cc}_{\bar{i}-1}}{\overline{X}^{cc}_{\bar{i}} - \overline{X}^{cc}_{\bar{i}-1}} \right) \bullet \left(\Delta\overline{X}^{c}_{\bar{i}} - \Delta\overline{X}^{c}_{\bar{i}-1} \right) \\
\bar{\xi}_{p} \geq 1/2 \;\; ; \;\; \Delta\overline{X}_{P} &= \Delta\overline{X}^{c}_{\bar{i}} + \left(\frac{\overline{X}_{p}(\bar{\xi}_{p}) - \overline{X}^{cc}_{\bar{i}}}{\overline{X}^{cc}_{\bar{i}+1} - \overline{X}^{cc}_{\bar{i}}} \right) \bullet \left(\Delta\overline{X}^{c}_{\bar{i}+1} - \Delta\overline{X}^{c}_{\bar{i}} \right)
\end{aligned} \tag{9}$$

Vertex-Based Methods The objective of vertex-based methods to describe the Jacobian matrix is to evaluate the coordinate transformation using vertex-centered control-volumes. Vertex-based methods inherently smooth the Jacobian matrix between cells; their control-volumes are defined by combining partial cell control-volumes. For rectilinear grids, one vertex-based method approximates the Jacobian matrix diagonal vector as the tracer-particles vertex control-volume widths, $\Delta\overline{X}^{v}_{\bar{i}+1/2}$. The resulting Jacobian matrix is constant throughout the vertex control-volume, but it is implicitly a function of the cell-based logical coordinates, $J = J(\overline{X}(\bar{\xi}))$. The vertex-based, piece-wise constant (Vrtx: Pwc) Jacobian matrix is presented in Equation 10.

$$\begin{aligned}
\bar{\xi}_{p} < 1/2 \;\; ; \;\; \Delta\overline{X}_{P} &= \Delta\overline{X}^{v}_{\bar{i}-1/2} \\
\bar{\xi}_{p} \geq 1/2 \;\; ; \;\; \Delta\overline{X}_{P} &= \Delta\overline{X}^{v}_{\bar{i}+1/2}
\end{aligned} \tag{10}$$

A vertex-based, piece-wise linear method for evaluating the coordinate transformation is

also possible for use within nonuniform rectilinear grids. This technique smooths out the discontinuity of $\Delta\overline{X}^{v}$ across vertex control-volumes. The Jacobian matrix is then smoothed twice; the matrix diagonals are inherently smoothed by using vertex control-volumes, and then they are smoothed again by interpolating between these values. The resulting Jacobian matrix is explicitly a function of the tracer-particle's logical coordinates, $J = J(\overline{X}(\overline{\xi}))$. The vertex-based, piece-wise linear (Vrtx: Pwl) Jacobian matrix is presented in Equation 11.

$$\Delta\overline{X}_P = (1-\overline{\xi}_P)\Delta\overline{X}^{v}_{i-1/2} + (\overline{\xi}_P)\Delta\overline{X}^{v}_{i+1/2} \tag{11}$$

Each of the above methods represents a practical technique of evaluating Jacobian matrices for nonuniform rectilinear grids. Their use within a logical-space advection technique is presented later in this article. When either Equations 5 or 6 are used for nonuniform rectilinear grids, only the cell-based, piece-wise constant Jacobian matrix predicts the correct logical coordinates in a single iteration. Each of the other Jacobian matrices requires multiple solution iterations even for these simple grids. In contrast, for uniform rectilinear grids, all of the above Jacobian matrices are identical and only one solution of Equation 5 or 6 is necessary to evaluate the correct tracer-particle logical coordinates.

Non-Orthogonal Grids

Around complex geometries, three-dimensional computational space is often discretized into nonorthogonal hexahedral cells. While the edges of these cells remain straight, their cell-faces are, in general, curved surfaces. For nonorthogonal cells, the spatial coordinates are coupled and the resulting Jacobian matrix is full as presented in Equation 12.

$$J = J(\overline{X}(\overline{\xi})) = \begin{bmatrix} \frac{\partial x}{\partial\overline{\xi}} & \frac{\partial x}{\partial\overline{\eta}} & \frac{\partial x}{\partial\overline{\zeta}} \\ \frac{\partial y}{\partial\overline{\xi}} & \frac{\partial y}{\partial\overline{\eta}} & \frac{\partial y}{\partial\overline{\zeta}} \\ \frac{\partial z}{\partial\overline{\xi}} & \frac{\partial z}{\partial\overline{\eta}} & \frac{\partial z}{\partial\overline{\zeta}} \end{bmatrix} \tag{12}$$

Trilinear interpolation is a general coordinate transformation method which is often used for hexahedral cells. This technique is used directly for data interpolation, and it may also be differentiated to construct Jacobian matrix elements. The resulting cell-based, trilinear Jacobian matrix (Cell: Tri) is appropriate for both orthogonal and nonorthogonal grids. For nonorthogonal grids, this matrix is explicitly a function of the logical coordinates, $J = J(\overline{X}(\overline{\xi}))$. In contrast, for uniform or nonuniform rectilinear grids, the trilinear Jacobian matrix is a constant diagonal

matrix and it is identical to the cell-based, piece-wise constant coordinate transformation.

Test Problems: Results and Discussion

Two test problems were selected for this study: physical-space and logical-space advection. Each test problem used a different, yet common advection technique; the objective was to evaluate their ability to predict tracer-particle pathlines within various grids. The evaluations involved computing pathlines from specified flow-fields, then comparing the predicted and exact solutions. Within each equation-of-motion, Equations 1 and 2, the physical velocity was computed directly from the known flow-field. The computational domain, a unit cube, was discretized with 3, 4, and 5 cells in the x, y, and z directions respectively. Within the structured grid system, a global CFL number of one was used to advance the tracer-particle position in time.

Physical-Space Advection

The physical-space advection technique solved Equation 1 using different time integration methods. The physical-space method of evaluating logical-coordinates, Equation 6, and the cell-based, trilinear Jacobian matrix were used throughout this algorithm. However, the logical-space method was used to predict new cell numbers within the localization process; it is the most straightforward method for structured grid systems. The resulting tracer-particle advection technique was similar to the one in Telluride, except the technique in Reference [19] was specialized for use in unstructured grid systems. Finally, for this test problem only, the interior grid vertices were randomly repositioned to create nonorthogonal cells.

A corner-flow velocity field, $\overline{V}_x = (3.8x, -3.8y, 0.8)^T$, was prescribed for the physical-space test problem, and the initial tracer-particle position was $\overline{X} = (0.02, 0.90, 0.10)^T$. The exact tracer-particle pathline for these conditions is presented in Figure 1. The predicted pathlines, using the standard Euler and the predictor-corrector integration methods, are presented in Figure 2. The vertical, linear coordinate of the pathline was correctly predicted using both integration methods; the vertical velocity component is uniform throughout the domain. The standard Euler method failed to correctly predict the nonlinear pathline coordinates which are products of the nonuniform velocity components. In contrast, the higher-order, predictor-corrector integration method correctly predicted the curved tracer-particle pathline within the nonorthogonal grid.

Logical-Space Advection

The logical-space advection technique solved Equation 2 using various Jacobian matrices. The pathlines were computed using both the standard Euler and predictor-corrector time integration methods. Within the localization process, logical-space methods were selected to both predict cell numbers and to evaluate the logical coordinates. To avoid an iterative solution of Equation 5, only the cell-based, piece-wise constant Jacobian matrix was used to evaluate the

logical coordinates. The resulting tracer-particle advection method closely matched the one in Pagosa, except the technique in Reference [10] was limited to a predictor-corrector integration method and used a different vertex-based, linearly interpolated Jacobian matrix.

A planar, uniform-flow velocity field, $\overline{V}_x = (0.0, 0.8, 0.8)^T$, was prescribed for the logical-space test problem, and the initial tracer-particle position was $\overline{X} = (0.10, 0.10, 0.10)^T$. The exact tracer-particle pathline for these conditions is presented in Figure 3. The predicted pathlines, computed within a uniform rectilinear grid, are presented in Figure 4. Though indistinguishable, results are presented for both the standard Euler and the predictor-corrector integration methods using all of the Jacobian matrices. The predicted pathlines are identical and correct using either integration method because the velocity flow-field is uniform throughout the domain. There is also no distinction between results using different coordinate transformations because the Jacobian matrices are identical for the uniform rectilinear grid.

The logical-space advection test problem was also computed on a nonuniform rectilinear grid with a cell-packing ratio of 1.2; along the coordinate axes, each cell edge was 20% larger then the previous cells width. The pathlines predicted using the standard Euler integration method are presented in Figure 5. In addition, Figures 6 and 7 present the difference between the predicted and exact solutions. The pathlines computed using the Cell:Tri and Cell:Pwc Jacobian matrices, which are identical and invariant within each orthogonal cell, match the exact solution. In contrast, the pathlines computed using the other Jacobian matrices are not well predicted.

The pathlines predicted using the predictor-corrector integration method within the nonuniform rectilinear grid are presented in Figure 8. In addition, Figures 9 and 10 present the difference between the predicted and exact solutions. The logical-space advection technique used for these solutions closely match the technique utilized in Reference [10] and the technique reported in Reference [11]. In Figure 8, none of the computed pathlines, using the various Jacobian matrices, match the correct tracer-particle trajectory across the entire domain. However, as shown in Figure 9 the y-coordinate of the pathlines computed using the constant Jacobian matrices, Cell:Tri and Cell:Pwc, match the correct position for the first few iterations. After the third time-step, these computed pathlines never recover the correct trajectory.

Within the logical-space advection results, three solutions predicted pathlines accurately: 1) uniform grid solutions using both the standard Euler and predictor-corrector integration method, and all of the Jacobian matrices, 2) nonuniform grid solutions using the standard Euler integration method and the constant Jacobian matrices, and 3) the first three y-coordinates of the nonuniform grid solutions using the predictor-corrector integration method and the constant

Jacobian matrices. (The third condition, which focuses on a single coordinate, is relevant here because the coordinate transformation matrix is uncoupled within the rectilinear grid.) All of these solutions share a common feature; the advecting Jacobian matrix, described below, is consistent with the initial or time-level ‘n’ Jacobian matrix.

The advecting Jacobian matrix, J^a, defined within the integration method, directly moves the tracer-particle through the logical velocity, $\overline{V}_{\bar{\xi}} = [J^a]^{-1}\overline{V}_x$. An initial Jacobian matrix is defined at the start of each time-step, $J^n = J(\overline{X}(\bar{\xi}^n))$. Coordinate transformation consistency, defined here as $J^a = J^n$, is guaranteed for the standard Euler integration method; the advecting logical velocity is computed at the initial position, $\overline{V}^n_{\bar{\xi}} = [J^n]^{-1}\overline{V}^n_x$. In contrast, Jacobian matrix consistency is not guaranteed for the predictor-corrector method; the advecting logical velocity is defined as $\overline{V}^{n+1/2}_{\bar{\xi}} = [J^{n+1/2}]^{-1}\overline{V}^{n+1/2}_x$. For this multi-step integration method, coordinate transformation consistency is guaranteed only if $J^a = J^{n+1/2} = J^n$, which is generally not true. When $J^a \neq J^n$ within the predictor-corrector integration method, Jacobian matrix consistency might still be obtained retroactively through modifications of Equation 5. However, this dependence of the integration method on a unique logical-coordinate evaluation method would violate the objective of modularity for the overall tracer-particle advection algorithm.

The logical-space advection results also imply that the advecting Jacobian matrix should only be a function of a single computational cell. Only pathlines predicted using the constant Jacobian matrices, Cell:Tri and Cell:Pwc, which are functions of one cell, matched the exact solution within the nonuniform grid. The other coordinate transformations anticipate cell width variations by averaging the Jacobian matrices. However, accurate pathlines were predicted using the averaged Jacobian matrices only within the uniform grid where they are identical to the constant coordinate transformations. Therefore, the results clearly show that an averaging methodology for coordinate transformations fails to capture the correct solution.

The predicted pathline errors, presented above, could have been reduced for the logical-space advection test problem. These solutions were computed on coarse grids with a relatively large cell-packing ratio, and the CFL number was a maximum for unsteady flows. However, tracer-particle pathlines are often computed in conjunction with more complex physical models. These more computationally intensive physics models typically dictate the required grid density and time-limit constraints. Therefore, the spatial and temporal accuracy of tracer-particle advection methods should be measured on relatively coarse grids with large time-steps.

Summary

The objective of this research was to investigate the ability of existing tracer-particle advection techniques to accurately predict pathlines within various grids. The advection algorithm was described as a collection of fundamental, yet independent components, and some of their common methods of implementation were presented. Two test problems were then selected for this study: physical-space and logical-space advection. Each test problem used a different, yet common technique to compute tracer-particle pathlines. The physical-space test problem, highly curved flow within a nonorthogonal grid, was fairly complex. The low-order integration method failed to match the exact solution. However, using a higher-order integration method, the physical-space advection technique accurately predicted the tracer-particle pathlines.

The logical-space test problem, planar uniform flow within a rectilinear grid, was very simple. While the pathlines predicted within the uniform grid were excellent, this technique produced surprisingly poor results within the nonuniform grid. However, the nonuniform grid solution did provide two insights into tracer-particle logical-space advection. First, only one Jacobian matrix should be used within the integration method for each time-step; the logical-space equation-of-motion similarly uses only a single coordinate transformation. Second, the Jacobian matrix should only be a function of a single computational cell; an averaging methodology for the coordinate transformation failed to accurately predict pathlines. Therefore, existing logical-space advection techniques are limited to uniform rectilinear grids which do not reflect the complex grids typically used for modern simulations.

References

1) Friedhoff, R. M. and Benzon, W., *The Second Computer Revolution: Visualization*, Harry N. Abrams, New York, 1989.

2) Silver, D., Zabusky, N., Fernandez, V., Gao, M., and Samtaney, R., "Ellipsoidal Quantification of Evolving Phenomena," *Scientific Visualization of Physical Phenomena*, Patrikalakis, N. M., Editor, Springer-Verlag, pp. 573-587, 1991.

3) Hockney, R. W. and Eastwood, J. W., *Computer Simulation Using Particles*, Hilger, Bristol, 1988.

4) *Ensight*, Computational Engineering International, Research Triangle Park, NC, 1998.

5) Pracht, W. E. and Brackbill, J. U., "BAAL: A Code for Calculating Three-Dimensional Fluid Flows at All Speeds with an Eulerian-Lagrangian Computing Mesh," Los Alamos National Laboratory Report LA-6342, 1976.

6) Murman, E. M. and Powell, K. G., "Trajectory Integration in Vortical Flows," *AIAA Journal*, Vol 27, pp. 982-984, 1989.

7) Fairfield, M. S., "Three-Dimensional Visualization of Reactive Flows in Complex Geometries," *High Performance Computing, Grand Challenges in Computer Simulation*, Tentner, A. M., Editor, Simulations Councils, pp. 424-427, 1995. (Los Alamos National Laboratory Report LAUR-95-0251, 1995.)

8) Cheng, H. P., Cheng, J. R., and Yeh, G. T., "A Particle Tracking Technique for the Lagrangian-Eulerian Finite Element Method in Multi-Dimensions," *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 1115-1136, 1996.

9) Darmofal, D. L. and Haimes, R., "An Analysis of 3D Particle Path Integration Algorithms," *Journal of Computational Physics*, Vol. 123, pp. 182-195, 1996.

10) Kothe, D. B., et al., "Pagosa: A Massively-Parallel, Multi-Material Hydrodynamics Model for Three-Dimensional High-Speed Flow and High-Rate Material Deformation," *High Performance Computing, Grand Challenges in Computer Simulation*, Tentner, A. M., Editor, Simulations Councils, pp. 9-14, 1993. (Los Alamos National Laboratory Report LAUR-92-4306, 1992.)

11) Shirayama, S., "Processing of Computed Vector Fields for Visualization," *Journal of Computational Physics*, Vol. 106, pp. 30-41, 1993.

12) Brock, J. S. and Painter, J. W., “Tracer-Particle Advection: Components, Choices, and a New Alternative,” Poster, Tenth NECDC Conference, Las Vegas, NV, 1998. (Los Alamos National Laboratory Report LAUR-98-4691, 1998.)

13) James, M. L., Smith, G. M., and Wolford, J. C., *Applied Numerical Methods for Digital Computation*, 3rd. ed., Harper & Row, New York, 1985.

14) Allievi, A. and Bermejo, R., “A Generalized Particle-Search Algorithm for Arbitrary Grids,” *Journal of Computational Physics*, Vol. 132, pp. 157-166, 1997.

15) Lohner, R., “Robust, Vectorized Search Algorithms for Interpolation on Unstructured Grids,” *Journal of Computational Physics*, Vol. 118, pp. 380-387, 1995.

16) Westerman, T., “Localization Schemes in 2D Boundary-Fitted Grids,” *Journal of Computational Physics*, Vol. 101, pp. 307-313, 1992.

17) Lohner, R. and Ambrosiano, J. “A Vectorized Particle Tracer for Unstructured Grids,” *Journal of Computational Physics*, Vol. 91, pp. 22-31, 1990.

18) Brackbill, J. U., and Ruppel, H. M., “FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions,” *Journal of Computational Physics*, Vol. 65, pp. 314-343, 1986.

19) Kothe, D. B., Ferrell, R. C., Turner, J. A., and Mosso, S. J., “A High Resolution Finite Volume Method for Efficient Parallel Simulation of Casting Processes on Unstructured Meshes,” Eighth SIAM Conference on Parallel Processing For Scientific Computing, Minneapolis, MN, 1997. (Los Alamos National Laboratory Report LAUR-97-0030, 1997).

Figures

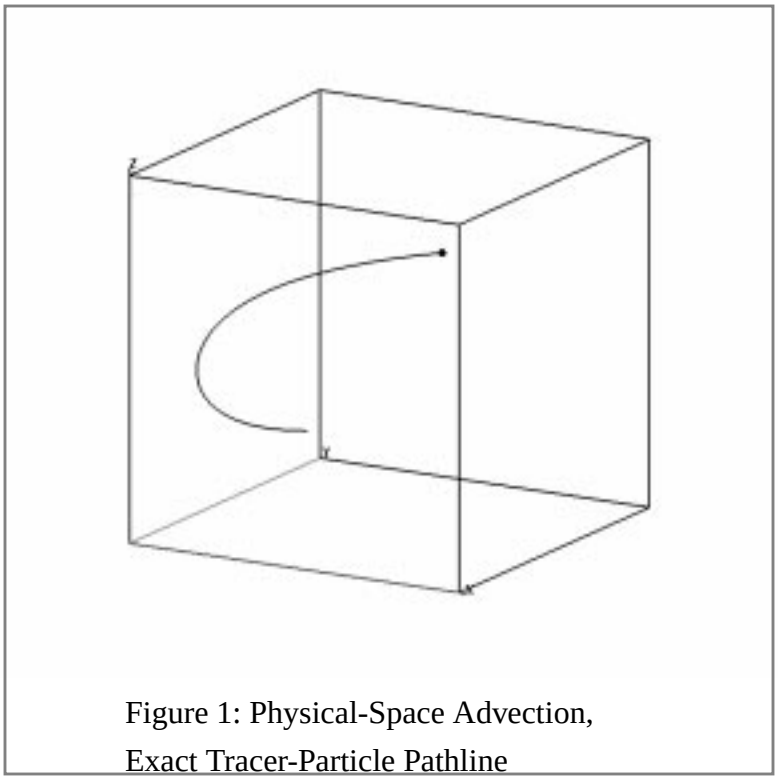

Figure 1: Physical-Space Advection,
Exact Tracer-Particle Pathline

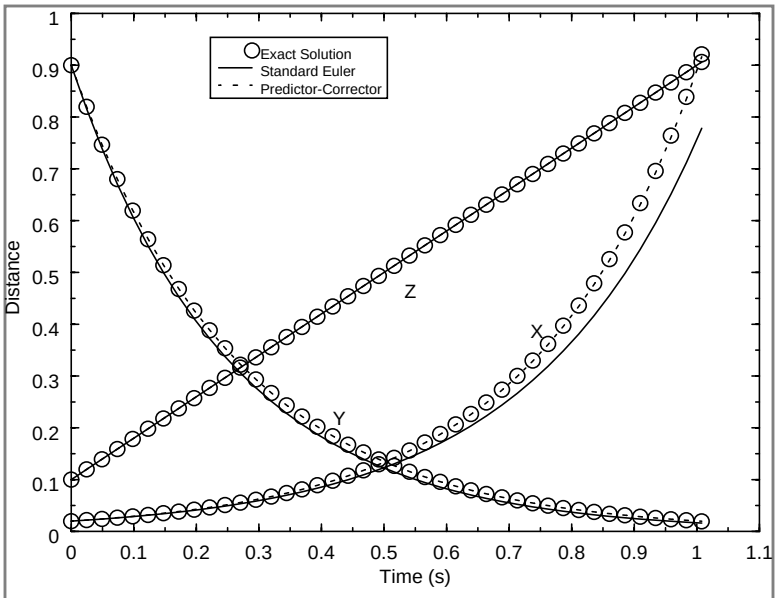

Figure 2: Physical-Space Advection,
Nonorthogonal Grid Solutions,
Standard Euler and Predictor-Corrector
Integration Methods

Figures Continued

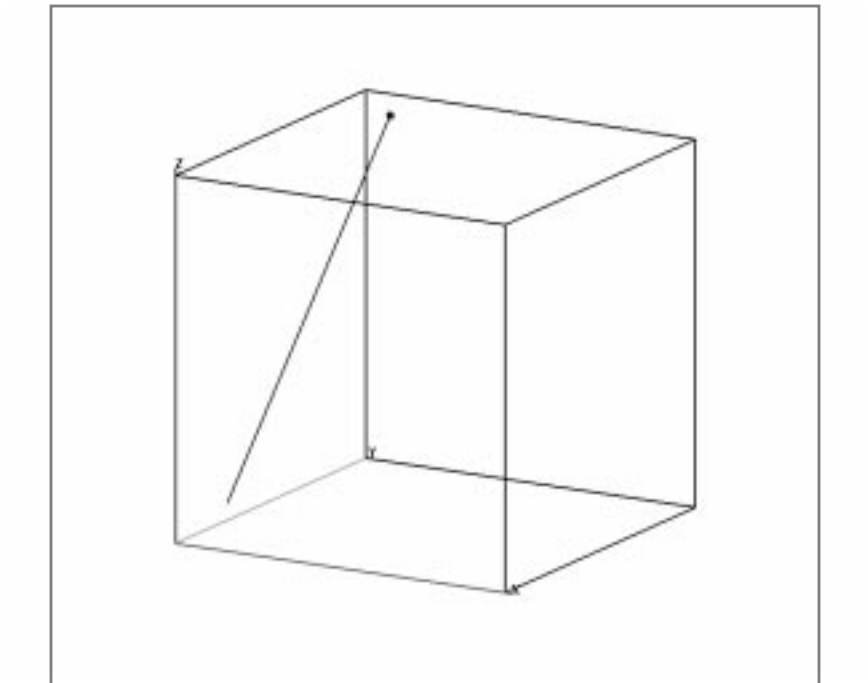

Figure 3: Logical-Space Advection,
Exact Tracer-Particle Pathline

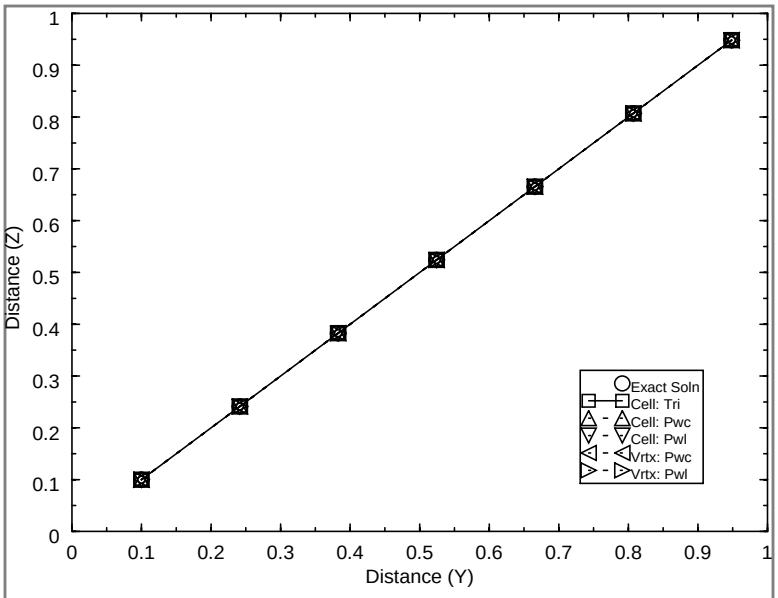

Figure 4: Logical-Space Advection,
Uniform Rectilinear Grid Solutions,
Standard Euler and Predictor-Corrector
Integration Methods

Figures Continued

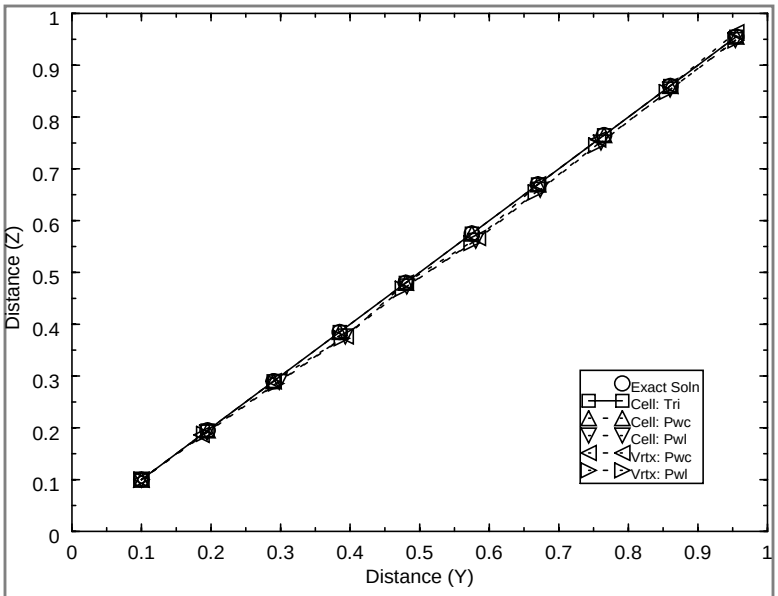

Figure 5: Logical-Space Advection, Nonuniform Rectilinear Grid Solutions, Standard Euler Integration Method

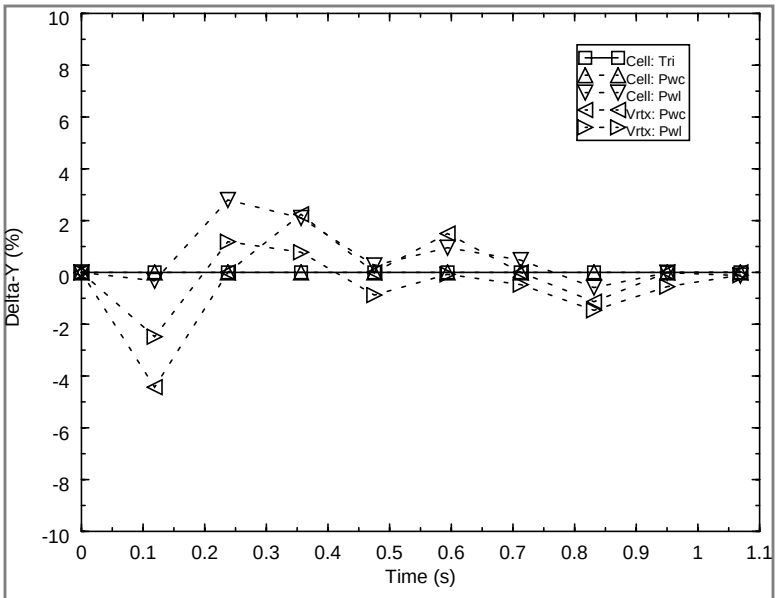

Figure 6: Logical-Space Advection, Nonuniform Rectilinear Grid Y-Deltas, Standard Euler Integration Method

Figures Continued

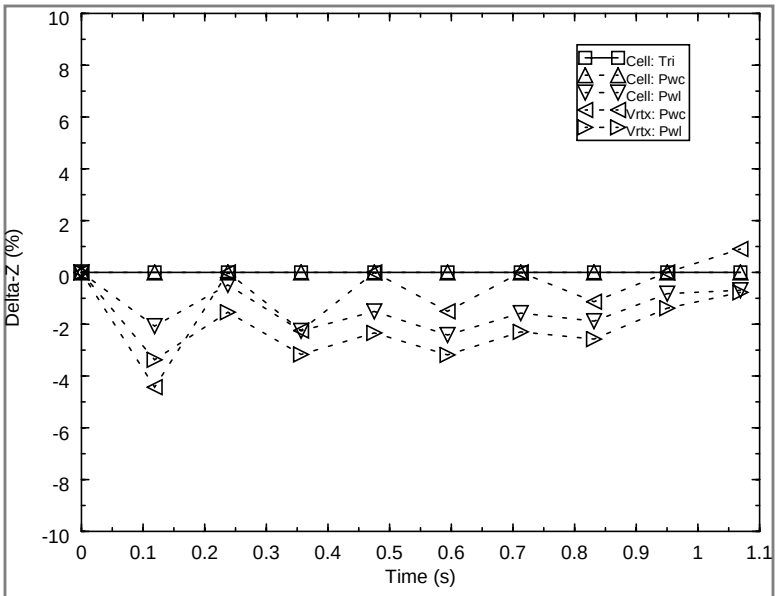

Figure 7: Logical-Space Advection, Nonuniform Rectilinear Grid Z-Deltas, Standard Euler Integration Method

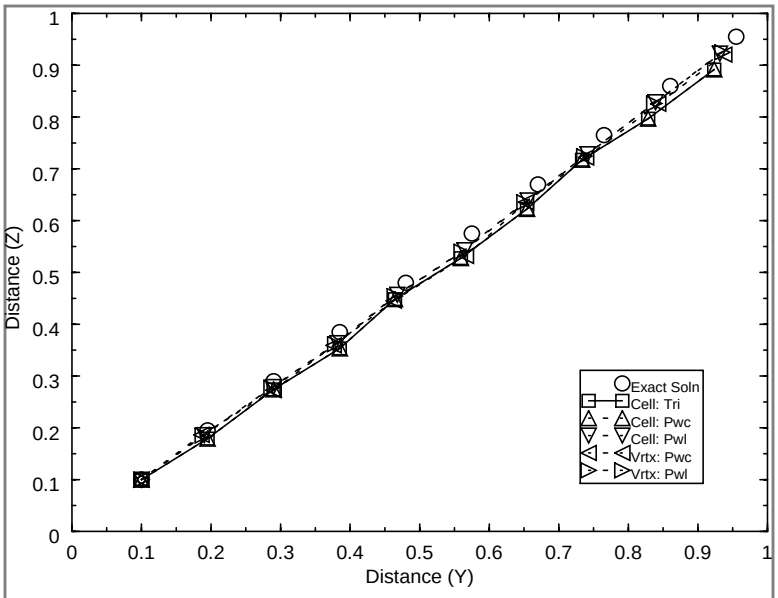

Figure 8: Logical-Space Advection, Nonuniform Rectilinear Grid Solutions, Predictor-Corrector Integration Method

Figures Continued

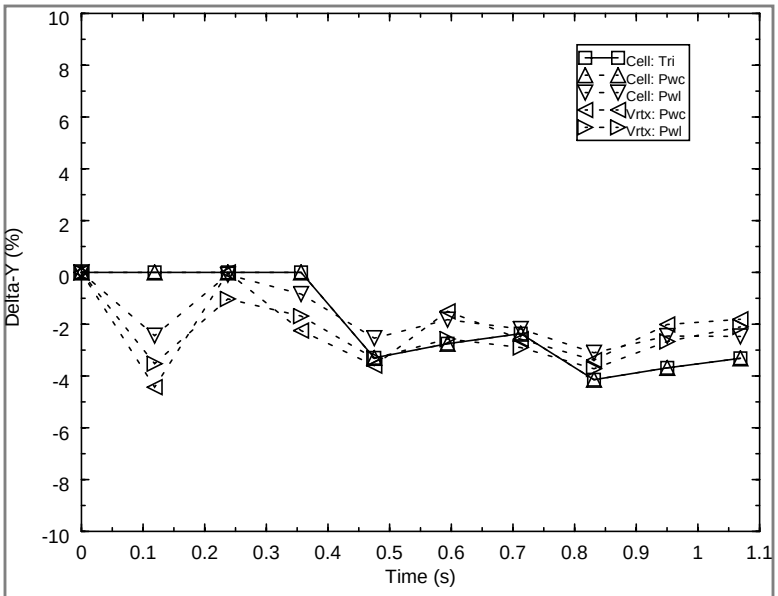

Figure 9: Logical-Space Advection, Nonuniform Rectilinear Grid Y-Deltas, Predictor-Corrector Integration Method

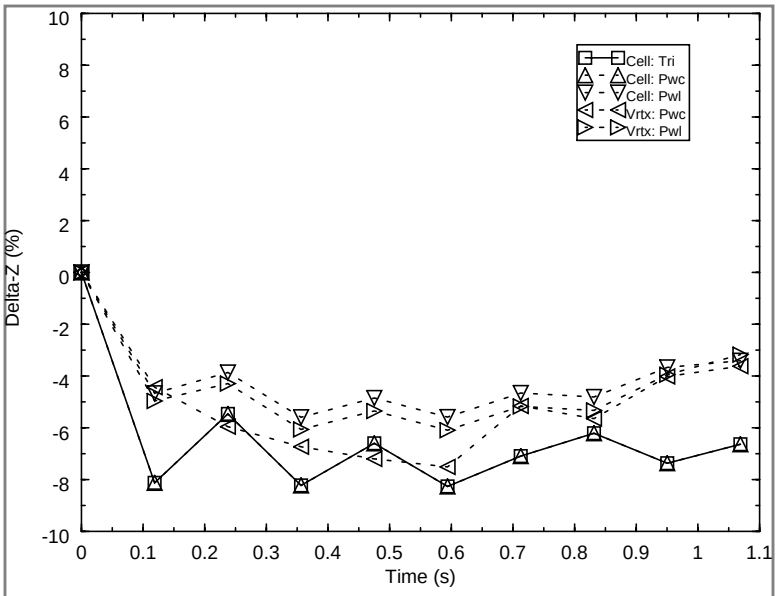

Figure 10: Logical-Space Advection, Nonuniform Rectilinear Grid Z-Deltas, Predictor-Corrector Integration Method